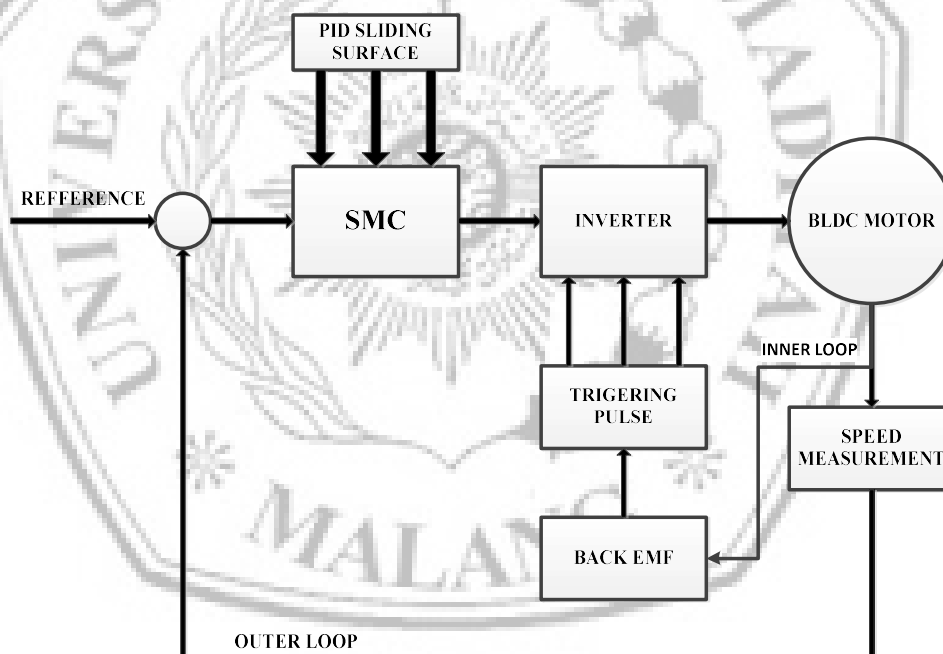


BAB III

PERANCANGAN DAN PEMBUATAN SISTEM

3.1. Perancangan Sistem

BLDC yang akan dikontrol akan diasumsikan sebagai system yang tidak stabil karena tujuan dari penelitian ini adalah untuk mengontrol ketidak stabilan system pada BLDC yang diakibatkan *durability* yang menurun karena pemakaian kerjanya. Salah satu hal yang mempengaruhi adalah panas yang diakibatkan dari pemakaian dalam jangka panjang. Panas yang ditimbulkan akan mengakibatkan nilai resistansi pada motor berubah sehingga sifat magnetis juga akan berubah. Oleh karena itu persamaan pada pemodelan akan dibuat dalam bentuk state space. Pemodelan yang akan dibuat pada perancangan sistem kendali akan digambarkan seperti diagram blok pada gambar 3.1.



Gambar 3.1 Skema Perancangan Sistem

3.1.1. Pengambilan Data Nilai Parameter Sistem

Data yang ditetapkan sebagai nilai dari parameter BLDCM pada penelitian ini merupakan data dari penelitian sebelumnya. Beberapa factor yang mempengaruhi kinerja sistem diabaikan seperti kerugian arus, perbedaan resistansi antar fase dan semikonduktor yang kurang ideal. Parameter yang

tertera pada Tabel 3.1 merupakan nilai yang digunakan pada pemodelan BLDC motor.

Tabel 3.1 Nilai Parameter Motor BLDC

Parameter	Symbol	Value	Units
Resistansi	R	0.25	Ohm
Induktansi	L	0.32	mH
Beda induktansi antar fasa	M	-1.5	mH
Konstanta <i>Back-EMF</i>	k_e	0.65	V/rad/s
Konstansa Torsi	k_t	0.0994	N-m/A
<i>Viscous Damping</i>	B	0.0096	N-m/(rad/s)
Momen inersia rotor	J	0.00042	Kg-m ²
Banyak kutub	P	8	<i>poles</i>

3.1.2. Perancangan BLDC Motor

Dalam penelitian dilakukan beberapa pemodelan sistem secara terpisah dimana model tersebut merupakan model dari komponen utama yang menunjang kinerja motor. Hal ini dilakukan karena komponen tersebut tidak dapat digambarkan dalam bentuk state space yang notabene merupakan pemodelan sistem utuh dari sebuah plant yang tidak memerlukan komponen penunjang.

Salah satu pemodelan yang sering digunakan untuk sistem tidak linier adalah dengan metode state space. State space merupakan metode yang mampu menggambarkan sistem dengan single input single output (SISO) dan juga multiple input multiple output (MIMO). Motor BLDC memiliki 3 fasa yang berbeda pada statornya, tetapi pada pemodelan akan digunakan 2 fasa yang mampu mewakili kinerja dan cara kerja motor tersebut dimana informasi dari fasa ketiga mampu didapatkan dari kedua fasa tersebut. Sehingga sistem akan tetap memiliki 3 fasa inpu dan output pada sistemnya.

Dari persamaan matematika yang telah digambarkan pada persamaan umum (2.1) dan (2.10) maka kita dapat membuat penyederhanaan sebagai berikut :

$$i_A + i_B + i_C = 0 \quad (3.1)$$

$$u_{AB} = r_a (i_a - i_B) + L_a \frac{d}{dt}(i_a - i_B) + e_{AB} \quad (3.2)$$

$$u_{BC} = r_a (i_a + 2i_B) + L_a \frac{d}{dt}(i_a + 2i_B) + e_{BC} \quad (3.3)$$

Jika persamaan (3.2) dan (3.3) di substitusikan maka :

$$u_{AB} - u_{BC} = -3 r_a i_B - 3L_a \frac{d}{dt}i_B + e_{AB} - e_{BC} \quad (3.4)$$

$$\frac{d}{dt}i_B = -\frac{r_a}{L_a} i_B - \frac{1}{3L_a}(u_{AB} - e_{AB}) + \frac{1}{3L_a}(u_{BC} - e_{BC}) \quad (3.5)$$

Dengan metode yang sama untuk mendapatkan turunan dari i_A maka bisa didapatkan dari persamaan :

$$u_{AB} = r_a (2i_a + i_B) + L_a \frac{d}{dt}(2i_a + i_B) + e_{AB} \quad (3.6)$$

$$u_{CA} = r_a (i_C - i_A) + L_a \frac{d}{dt}(i_C - i_A) + e_{CA} \quad (3.7)$$

Persamaan (3.6) dan (3.7) disubstitusikan menjadi :

$$(u_{AB} - e_{AB}) - (u_{CA} - e_{CA}) = -3 r_a i_A - 3L_a \frac{d}{dt}i_A \quad (3.8)$$

$$\frac{d}{dt}i_A = -\frac{r_a}{L_a} i_A + \frac{1}{3L_a}(u_{AB} - e_{AB}) - \frac{1}{3L_a}(u_{CA} - e_{CA}) \quad (3.9)$$

$$u_{AB} = u_{BC} \quad (3.10)$$

$$e_{AB} = e_{BC} \quad (3.11)$$

$$u_{CA} = -(u_{AB} + u_{BC}) = -2 u_{AB} \quad (3.12)$$

$$e_{CA} = -(e_{AB} + e_{BC}) = -2 e_{AB} \quad (3.13)$$

Ketika persamaan (3.10), (3.11), (3.12), (3.13), di substitusikan pada persamaan (3.9) maka akan menghasilkan persamaan sebagai berikut :

$$\frac{d}{dt}i_A = -\frac{r_a}{L_a} i_A + \frac{1}{3L_a}(u_{BC} - e_{BC}) + \frac{2}{3L_a}(u_{AB} - e_{AB}) \quad (3.14)$$

$$r_a = 2R \quad (3.15)$$

$$L_a = L - M \quad (3.16)$$

Pada persamaan (2.10) dapat dinyatakan dalam bentuk lain yaitu :

$$\dot{\omega} = -\frac{B}{J}\omega + \frac{1}{J}(T_e - T_l) \quad (3.17)$$

Dimana T_l merupakan beban dan gangguan sedangkan T_e adalah torsi yang dihasilkan oleh motor yang dapat digambarkan dengan persamaan :

$$T_e = K_t i \quad (3.18)$$

Dengan Persamaan (3.5), (3.14) dan (3.17) maka akan ditemukan persamaan dinamis yang akan dibuat dalam bentuk state space dengan pola

$$\begin{aligned} \dot{x} &= Ax + Bu + Gw \\ y &= Cx + Du \end{aligned} \quad (3.19)$$

Sehingga menghasilkan matrix state space sebagai berikut:

$$\begin{bmatrix} \dot{ia} \\ \dot{ib} \\ \dot{\Omega} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{-ra}{La} & 0 & 0 & 0 \\ 0 & \frac{-ra}{La} & 0 & 0 \\ 0 & 0 & \frac{-Bv}{J} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} ia \\ ib \\ \Omega \\ \theta \end{bmatrix} + \begin{bmatrix} \frac{2}{3La} & \frac{2}{3L} & 0 \\ \frac{-1}{3La} & \frac{1}{3La} & 0 \\ 0 & 0 & \frac{1}{J} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_{AB} - e_{AB} \\ u_{BC} - e_{BC} \\ Te - Tl \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\frac{Fc}{J} \end{bmatrix} \text{sgn}(\Omega)$$

$$\begin{bmatrix} \dot{ia} \\ \dot{ib} \\ \dot{ic} \\ \dot{\omega r} \\ \dot{\theta r} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} ia \\ ib \\ ic \\ \omega r \\ \theta r \end{bmatrix} + [0]d$$

Dimana F_c merupakan koefisien *Coulomb Friction*. Dan koefisien itu berlawanan dengan gaya inersia J yang menyebabkan gerak putar motor melambat. Koefisien *Coulomb Friction* memiliki fungsi menghambat kerja motor dan membuat kerja motor menjadi tidak linear.

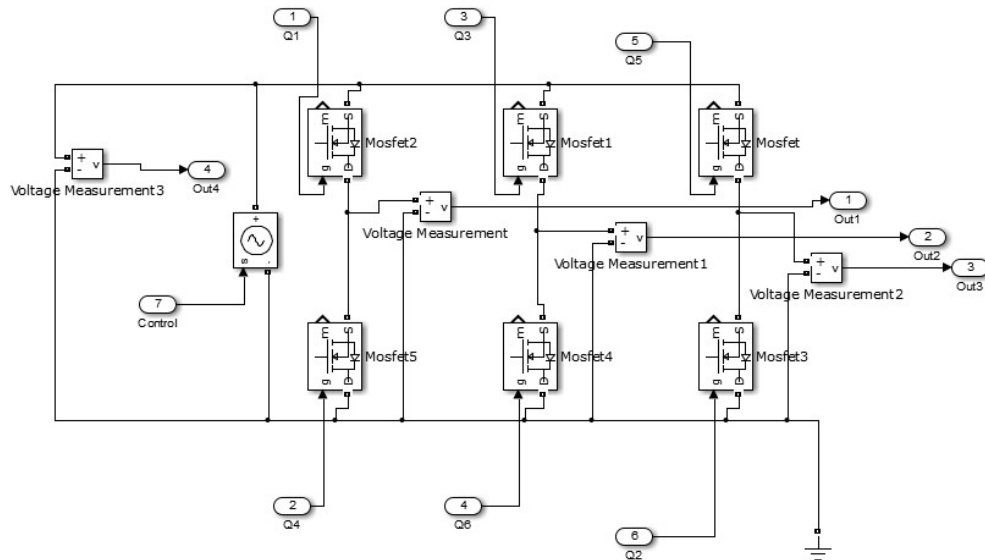
3.1.3. Pemodelan Driver Motor

Driver motor merupakan komponen paling penting sebagai penunjang bekerjanya motor. Driver motor BLDC bekerja seperti layaknya inverter pada motor AC 3 fasa, dikarenakan karakteristik BLDCM yang hampir sama dengan motor AC 3 fasa. Arus DC yang diberikan oleh sumber kepada motor akan di substitusikan menjadi 3 fasa yang memiliki perbedaan sudut. Pemodelan ini menggunakan power electronic library yang terdapat pada Matlab dengan struktur penyalan dan skema kerja sebagai berikut :

Tabel 3.2 Tabel Penyalan Transistor pada Driver Motor

	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
Transistor 1	0	0	1	1	0	0
Transistor 4	1	0	0	0	0	1
Transistor 3	0	0	0	0	1	1
Transistor 6	0	1	1	0	0	0
Transistor 5	1	1	0	0	0	0
Transistor 2	0	0	0	1	1	0

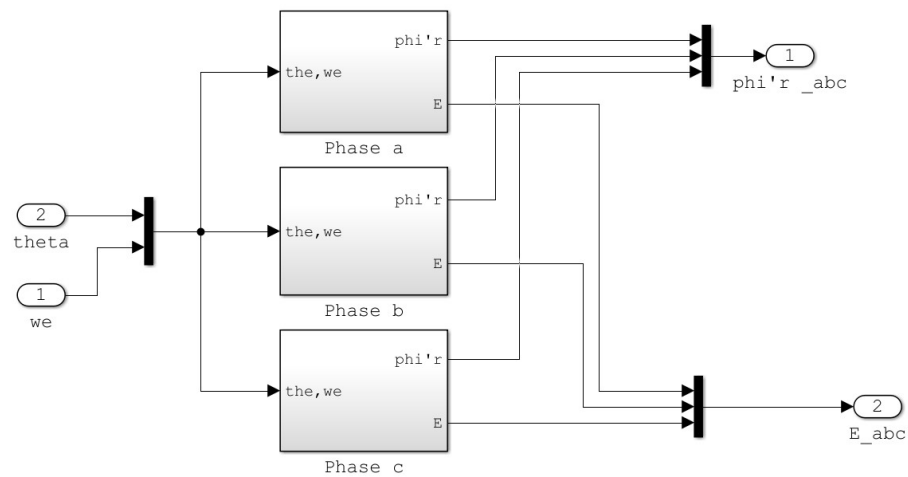
Representasi dari table penyalan thyristor dimodelkan pada matlab seperti gambar 3.1



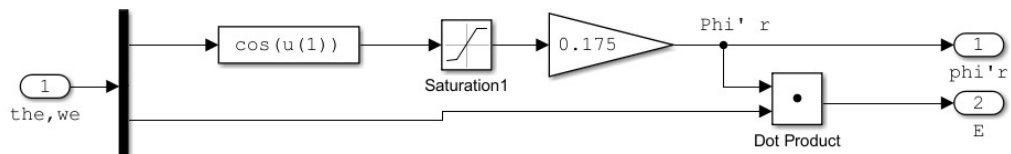
Gambar 3.2 Model Driver Motor

3.1.4. Pemodelan Back EMF

Pada model dinamis dari motor terdapat fungsi dari back EMF yang tidak dapat dideskripsikan dalam bentuk state space. Persamaan back EMF merupakan persamaan trapezoidal yang memiliki 3 fasa yang berbeda. Pada pemodelan ini diasumsikan bahwa keluaran back EMF dari motor bersifat simetris sehingga motor berputar secara ideal. Back EMF memiliki sinyal yang pergeserannya berbeda 120 derajat setiap fasanya dan dimodelkan pada Matlab seperti gambar 3.2 dan 3.3.



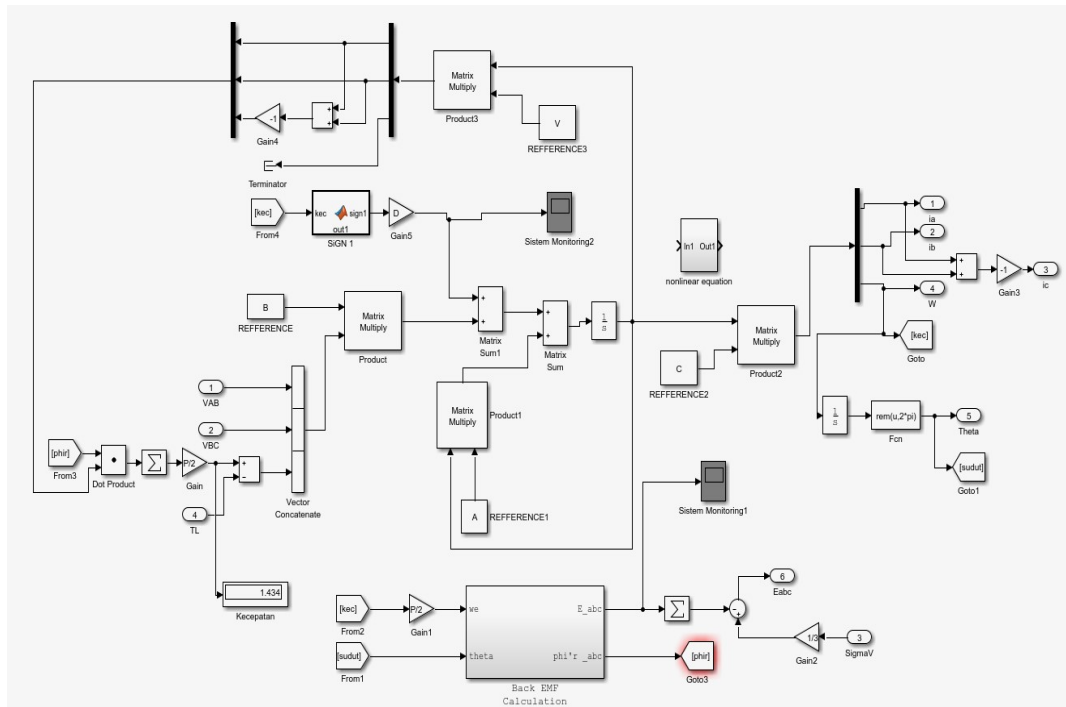
Gambar 3.3 Model Back EMF



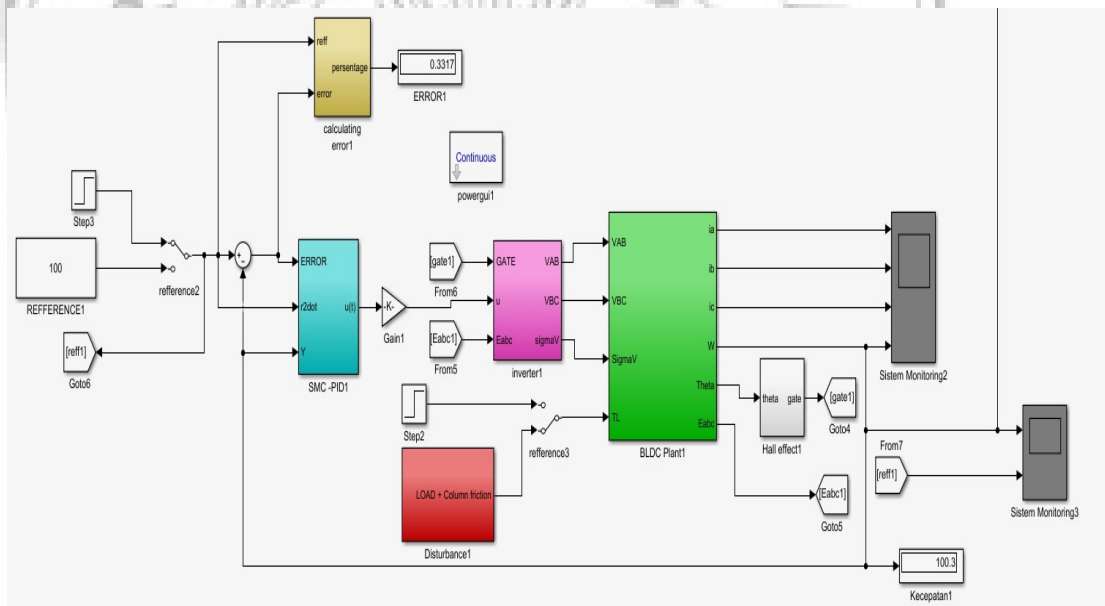
Gambar 3.4 Pembentukan Trapezoidal pada Back EMF

3.1.5. Pemodelan Motor BLDC

Dengan terbentuknya model dari beberapa komponen utama dari BLDC maka persamaan dinamis yang dimiliki oleh motor pun dapat dimodelkan. Pemodelan dari state space serta komponen utama motor akan menggambarkan kinerja motor sesuai dengan keadaan aslinya, dimana karakteristik kelistrikan dan mekanisnya dapat disimulasikan pada Simulink dengan skema seperti gambar 3.4 dan 3.5.



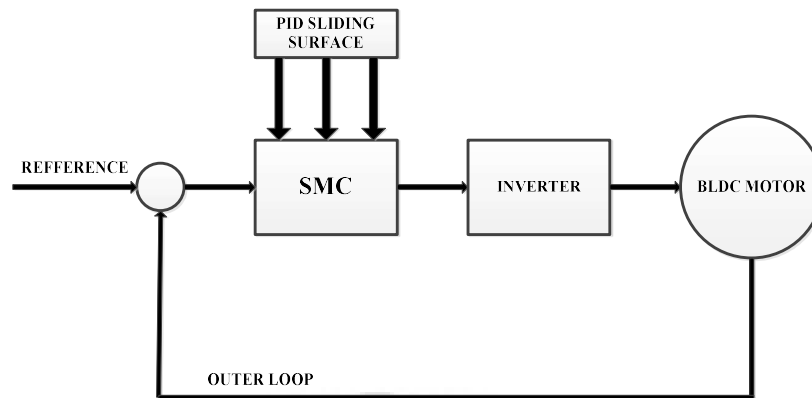
Gambar 3.5 Model Dinamis Motor BLDC



Gambar 3.6 Model Motor BLDC pada sistem close loop dengan driver motor

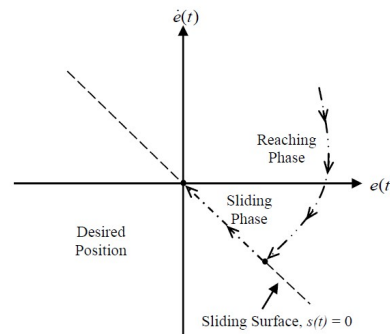
3.2. Perancangan SMC dengan PID *Sliding Surface*

Implementasi dari SMC dengan PID Sliding surface akan dimodelkan seperti gambar blok diagram berikut :



Gambar 3.7 Blok Diagram Konsep Metode Pengontrolan dengan SMC-PID

Desain tersebut diharapkan dapat mengurangi pengaruh dari gangguan yang tidak diinginkan diantaranya perubahan panas dan gesekan. Setelah melakukan pemodelan BLDC, PID-GA akan ditambahkan pada system meminimalisir kesalahan kecepatan putar antara output motor dan nilai referensi, dimana nilai K_i , K_p , dan K_d akan di hitung menggunakan algoritma genetika. Kemudian SMC akan di terapkan pada system dimana PID tersebut akan terintegrasi sebagai sliding surface dari controller SMC dan pemantauan performa system akan dilakukan. Struktur SMC dipaparkan dengan gambar 3.9.



Gambar 3.8 Struktur Pada Perancangan Sliding Mode Control

PID digunakan sebagai sliding surface dari SMC dengan nilai K_p , K_i dan K_d sebagai salah satu koefisien dari perhitungan SMC. Persamaan matematikanya adalah sebagai berikut:

$$S(t) = K_p e(t) + K_i \int e(t) + K_d \frac{de}{dt} \quad (3.20)$$

Error yang terjadi akibat perbedaan antara nilai actual dengan trajectory yang diinginkan digambarkan dengan persamaan:

$$e(t) = r(t) - y(t) \quad (3.21)$$

Dimana $e(t)$ merupakan error yang didapatkan dari selisih nilai refferensi $r(t)$ dan nilai output sistem $y(t)$. Jika persamaan menggunakan orde 2 pada modelnya maka dapat dituliskan pada persamaan:

$$\ddot{e}(t) = \ddot{r}(t) - \ddot{y}(t) \quad (3.22)$$

$$\ddot{y}(t) = \ddot{\omega} = \ddot{\Omega}$$

Dari persamaan (3.2), (3.17) dan (3.18), $\ddot{\omega}$ dapat digambarkan sebagai berikut:

$$\ddot{\omega} = -A\dot{\omega} - B\omega + Cu(t) - F(t) \quad (3.23)$$

$$u(t) = v(t) \quad (3.24)$$

$$A = -\frac{B}{J} \quad (3.25)$$

$$B = -\frac{Kt^2}{JLa} \quad (3.26)$$

$$C = -\frac{Kt}{JLa} \quad (3.27)$$

$$F(t) = Te - Tl \quad (3.28)$$

Pada umumnya SMC digambarkan memiliki persamaan yang menjumlahkan antara switching control dan equivalen control. Switching control merupakan persamaan penyesuaian saat nilai $s(t) \neq 0$ yang dilambangkan dengan symbol u_{sw} dan equivalen control adalah persamaan penyesuaian saat $s(t) = 0$. Sehingga U_{SMC} adalah :

$$U_{SMC} = U_{eq} + U_{sw} \quad (3.29)$$

Karena nilai dari $s(t) = 0$ maka dapat diasumsikan bahwa $s(t) = \dot{s}(t) = 0$ maka dapat digambarkan turunan kedua persamaan $s(t)$ menjadi :

$$\dot{s}(t) = K_p \dot{e}(t) + K_i e(t) + K_d \ddot{e}(t) \quad (3.30)$$

Kemudian diasumsikan juga bahwa nilai dari beban adalah 0, Dan persamaan (3.23) dimasukan sebagai pengganti nilai $\ddot{e}(t)$ maka dapat ditulis persamaan equivalen controlnya adalah:

$$\dot{s}(t) = K_p \dot{e}(t) + K_i e(t) + K_d (\ddot{r}(t) + A\dot{\omega} + B\omega - Cu(t)) \quad (3.31)$$

Saat $\dot{s}(t) = 0$ dapat dinyatakan bahwa persamaan equivalen SMC adalah:

$$u_{eq} = (K_d C)^{-1} (K_p \dot{e}(t) + K_i e(t) + K_d (\ddot{r}(t) + A\dot{\omega} + B\omega)) \quad (3.32)$$

Dari teori analisa kestabilan yaitu teori Lyapunov yang telah di verifikasi kestabilannya, controller ini akan mengikuti fungsi Lyapunov yaitu:

$$V = \frac{1}{2}s^2 \quad (3.34)$$

Dimana $V(t) > 0$ dan $V(0) = 0$ untuk $s(t) \neq 0$. Kondisi pencapaian akan tergambar pada persamaan

$$\dot{V}(t) = s(t) \dot{s}(t) < 0 \quad ; s(t) \neq 0 \quad (3.35)$$

Persamaan ini memiliki tujuan untuk memastikan nilai bergerak dari kondisi pencapaian menuju ke sliding phase pada kondisi stabil. $\dot{V}(t)$ harus dipastikan negatif untuk parameter SMC dimana parameter tersebut akan terisi saat system berada pada kondisi pencapaian atau reaching phase.

$$0 > s\dot{s}$$

$$\begin{aligned} 0 &> s(Kp \dot{e}(t) + Ki e(t) + Kd (\ddot{r}(t) + A\dot{\omega} + B\omega - Cu(t))) \\ 0 &> s(Kp \dot{e}(t) + Ki e(t) + Kd (\ddot{r}(t) + A\dot{\omega} + B\omega) - Kd Cu(t)) \\ 0 &> s(Kp \dot{e}(t) + Ki e(t) + Kd (\ddot{r}(t) + A\dot{\omega} + B\omega) - Kd C(ueq + usw)) \\ 0 &> s(Kp \dot{e}(t) + Ki e(t) + Kd (\ddot{r}(t) + A\dot{\omega} + B\omega) - Kd C(KdC)^{-1}(Kp \dot{e}(t) + \\ &Ki e(t) + Kd(\ddot{r}(t) + A\dot{\omega} + B\omega)) + Kd C u_{sw})) \\ 0 &> s(Kp \dot{e}(t) + Ki e(t) + Kd (\ddot{r}(t) + A\dot{\omega} + B\omega)) - sKd C(KdC)^{-1}(Kp \dot{e}(t) + \\ &Ki e(t) + Kd(\ddot{r}(t) + A\dot{\omega} + B\omega)) + sKd C u_{sw})) \end{aligned} \quad (3.36)$$

Untuk memastikan bahwa persamaan (3.36) bernilai kurang dari nol maka di tentukan nilai u_{sw} adalah

$$\begin{aligned} U_{sw} = & \text{sign}(s)(Kp \dot{e}(t) + Ki e(t) + Kd (\ddot{r}(t) + A\dot{\omega} + B\omega) \\ & - Kd C(KdC)^{-1}(Kp \dot{e}(t) + Ki e(t) + Kd(\ddot{r}(t) + A\dot{\omega} + B\omega))) \end{aligned} \quad (3.37)$$

Dengan diterapkannya sign function pada sliding surface sehingga akan didapatkan persamaan untuk switching controlnya yaitu :

$$U_{sw} = Ks \text{sat}\left(\frac{s}{\varphi}\right) \quad (3.38)$$

$$\begin{aligned} \text{sat}\left(\frac{s}{\varphi}\right) &= \left(\frac{s}{\varphi}\right) & ; \text{if } \left|\frac{s}{\varphi}\right| \leq 1 \\ \text{sat}\left(\frac{s}{\varphi}\right) &= \text{sign}\left(\frac{s}{\varphi}\right) & ; \text{if } \left|\frac{s}{\varphi}\right| > 1 \end{aligned} \quad (3.39)$$

Persamaan diatas pada prakteknya memiliki cara kerja seperti prediksi control relay ideal. Jadi dalam kasus ini persamaan sign bisa digantikan dengan fungsi tangen hiperbolik untuk meningkatkan kinerja *hitting control*. Dimana

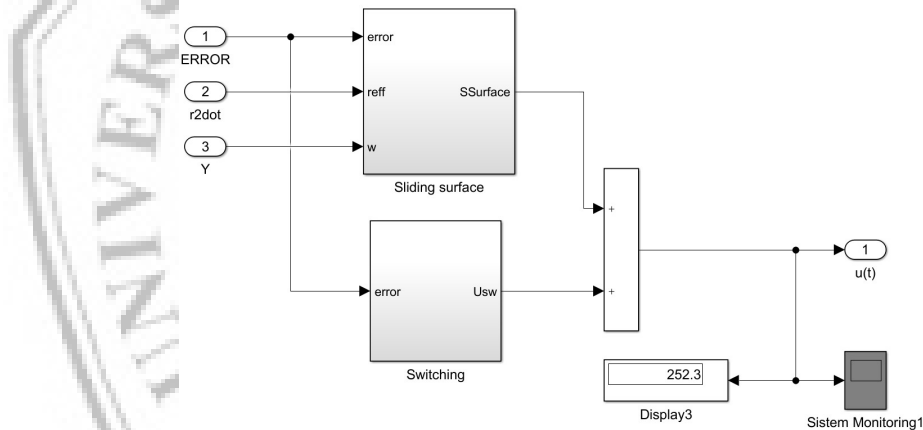
nilai K_s merupakan penguatan switching control dan φ merupakan ketebalan dari *boundary layer*. sehingga persamaan switching dapat dituliskan:

$$U_{sw} = K_s \tanh\left(\frac{s}{\varphi}\right) \quad (3.40)$$

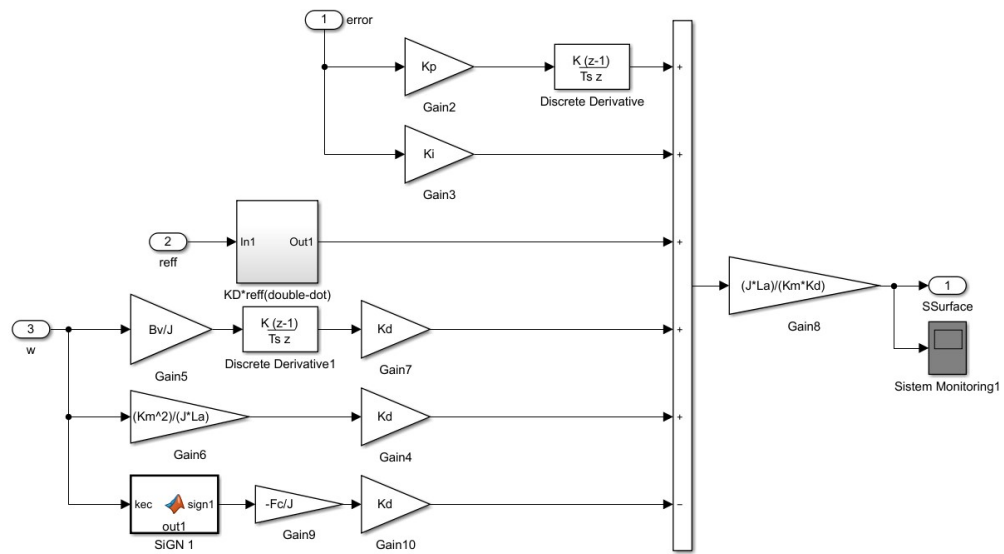
Untuk memastikan kestabilan system dari switching control yang berdasar kepada teori Lyapunov, beragam efek dari fungsi *discontinuous* telah di kurangi dengan menggantinya ke fungsi tangen hiperbolik dengan lapisan pembatas dari φ sebagai pendekatan. Dan hasil dari U_{smc} pada persamaan (3.32) dan (3.35) yang disubstitusikan pada persamaan (3.29) adalah sebagai berikut.

$$u_{smc} = (KdC)^{-1}(Kp \dot{e}(t) + Ki e(t) + Kd (\ddot{r}(t) + A\dot{\omega} + B\omega)) + K_s \tanh\left(\frac{s}{\varphi}\right) \quad (3.41)$$

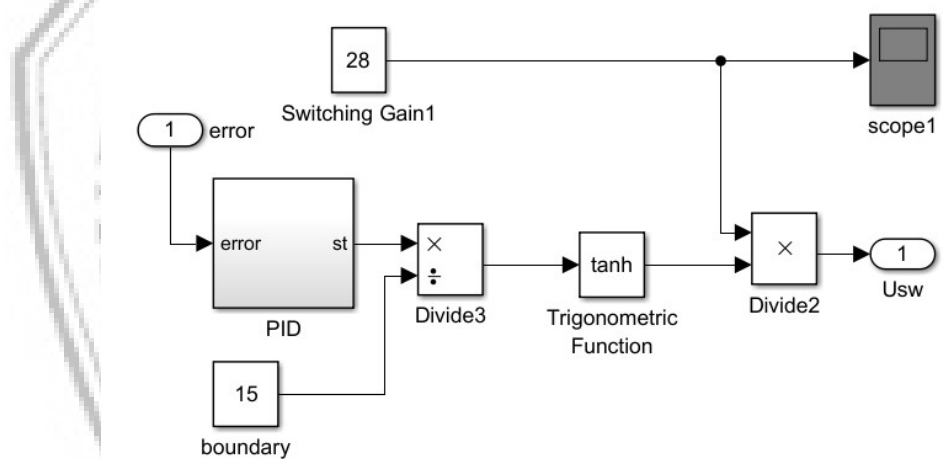
Dari persamaan 3.41 maka disimulasikan sistem control SMC pada Simulink sebagai berikut :



Gambar 3.9 Pemodelan SMC dengan PID sliding surface



Gambar 3.10 Pemodelan Ueq PID sliding surface

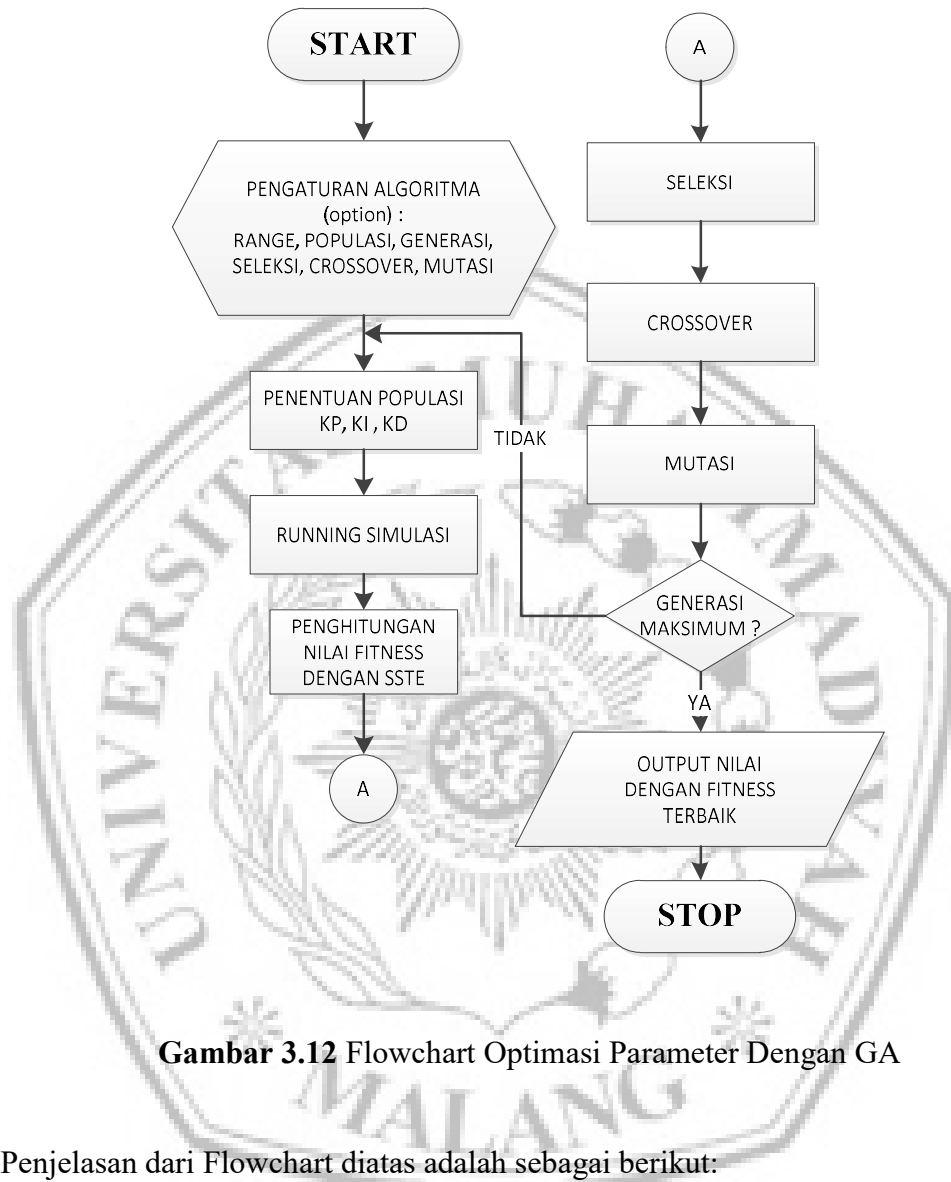


Gambar 3.11 Pemodelan Usw sebagai switching

3.3. Penentuan Parameter PID *Sliding Surface* Dengan Metode Algoritma Genetika

Pada persamaan (3.41) terdapat parameter K_p , K_i , dan K_d dimana nilainya akan menentukan respon transien sistem yang akan dikontrol, sehingga nilainya harus benar benar sesuai untuk plant tersebut. Untuk mendapat nilai yang paling optimal maka dilakukan optimasi dengan menggunakan algoritma genetika. Nilai parameter menggambarkan kromosom x yang memiliki 3 individu $x = [K_p \ K_i \ K_d]$. Kemudian dilakukan pengaturan algoritma yaitu ukuran populasi, generasi,

jenis crossover dan jenis mutasi. Optimalisasi dilakukan dengan Matlab editor yang memiliki flowchart sebagai berikut :



Gambar 3.12 Flowchart Optimasi Parameter Dengan GA

Penjelasan dari Flowchart diatas adalah sebagai berikut:

1. Option

Perintah ini berfungsi untuk melakukan pengaturan awal dari optimasi. Didalam perintah ini terdapat beberapa pengaturan dan juga nilai parameter pengaturan yang nantinya akan dilakukan oleh algoritma genetika saat melakukan optimalisasi kromosomnya.

2. PopInitRange

PopInitRange merupakan pengaturan untuk mengatur jarak populasi antar individu. Nilai pembatasan pembangkitan populasi dilakukan diantara nilai

optimal yang sebelumnya dicari dengan metode *trial and error*, hal ini dilakukan untuk memastikan nilai individu tetap pada kondisi stabil sistem.

3. PopulationSize

Penentuan jumlah populasi pada pengaturan ini dilakukan sesuai dengan kebutuhan optimalisasi. Pada kasus ini hanya digunakan 50 populasi karena nilai ini sudah cukup menghasilkan solusi nilai yang dioptimalkan.

4. Generation

Jumlah generasi adalah iterasi yang akan dilakukan dalam proses perhitungan. Makin besar nilai generasi maka akan menghasilkan nilai yang lebih baik dalam optimalisasi tetapi dengan banyaknya nilai generasi maka waktu yang diperlukan untuk melakukan optimalisasi semakin lama.

5. CrossoverFcn

Perintah ini digunakan untuk memilih jenis crossover yang akan dilakukan oleh algoritma dalam perhitungan. Dalam hal ini dipilih cross over scattered yang merupakan pengaturan *default* dari GA, dimana penyilangan tidak tergantung pada nilai induknya dan dilakukan penggabungan secara acak.

6. SelectionFcn

Fungsi seleksi adalah untuk menentukan individu yang akan digunakan untuk pembangkitan populasi berikutnya. Dipilihnya selection tournament karena metode ini menggunakan nilai tertinggi dalam setiap pembangkitan populasi yang lebih efisien dari pada penentuan secara acak.

7. MutationFcn

Mutasi dilakukan dengan metode mutation uniform karena pemilihan individu yang digunakan berdasarkan pada nilai probabilitas pada setiap generasi. Probabilitas pada mutasi ini juga akan mengendalikan nilai gen baru yang akan dimunculkan.

8. [x] = ga(@tuningPIDsmc)

Perintah ini digunakan untuk mengeksekusi perhitungan yang sudah diatur sebelumnya pada perintah option. Pada perintah ini juga disertakan fungsi yang akan dioptimalkan oleh GA. Fungsi fitness harus berada di bawah bagan perintah ini agar proses eksekusi bisa dimulai.

3.4. Luenberger Observer

Sistem linier yang menggunakan sistem control pasti memiliki umpan balik untuk dibandingkan dengan nilai referensi yang kemudian akan menjadi input bagi sistem control tersebut. Umpan balik dari sistem bisa diatur sesuai dengan refferensi apa yang ingin diberikan. Tetapi untuk memastikan bahwa keluaran umpan balik sesuai dengan state pada sistem maka dibutuhkan sebuah *disturbance observer* salah satunya adalah *Luenberger Observer* [7]. Permasalahan yang muncul pada control SMC adalah adanya chattering dimana nilainya harus disesuaikan pada gangguan terbesar yang akan diterima oleh motor. Hal ini menyebabkan sistem akan melakukan chattering yang besar walaupun saat terjadi gangguan kecil. Oleh karena itu digunakan Observer yang berfungsi untuk mendeteksi gangguan pada motor kemudian dikonversikan menjadi nilai penguatan switching (Ks) pada SMC sehingga nilai chattering akan beradaptasi sesuai dengan nilai gangguan yang diterima. *Luenberger Observer* memiliki persamaan dasar yaitu[12] :

$$\dot{\hat{x}} = A\hat{x} + Bu + L[y - C\hat{x}] \quad (3.42)$$

Dengan estimasi error observer adalah :

$$\hat{e} = \hat{x} - x \quad (3.43)$$

$$\dot{\hat{e}} = (A - LC)\hat{e} - Ed \quad (3.44)$$

$$A - LC = \begin{bmatrix} \frac{-ra}{La} & 0 & 0 \\ 0 & \frac{-ra}{La} & 0 \\ 0 & 0 & \frac{-Bv}{J} \end{bmatrix} - \begin{bmatrix} L1 \\ L2 \\ L3 \end{bmatrix} \cdot [0 \ 0 \ 1] \quad (3.45)$$

$$\lambda I - (A - LC) = \lambda \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} \frac{-ra}{La} & 0 & -L1 \\ 0 & \frac{-ra}{La} & -L2 \\ 0 & 0 & \frac{-Bv}{J} - L3 \end{bmatrix} \quad (3.46)$$

Nilai karakteristik polynomial pada Luenberger Observer adalah determinan dari matrix (48).

$$\begin{aligned}
& \det(\lambda I - (A - LC)) \\
&= \lambda^3 + \lambda^2(-2\rho - \sigma + L3) + \lambda(2\rho\sigma + \rho^2 - 2L3\rho) \\
&+ (-\rho^2\sigma + \rho^2L3)
\end{aligned} \tag{3.47}$$

Dimana

$$\rho = -ra/La \tag{3.48}$$

$$\sigma = -Bv/J \tag{3.49}$$

Menurut hukum persamaan Vieta, nilai akar- akar pada persamaan pangkat 3 adalah

$$-(\lambda_1 \lambda_2 \lambda_3) = -\rho^2\sigma + \rho^2L3 \tag{3.50}$$

$$L3 = \frac{-(\lambda_1 \lambda_2 \lambda_3) + \rho^2\sigma}{\rho^2} \tag{3.51}$$

Sehingga didapatkan Nilai L pada state Observer

$$L = \begin{bmatrix} 0 \\ 0 \\ \frac{-\lambda_1\lambda_2\lambda_3 + \rho^2\sigma}{\rho^2} \end{bmatrix} \tag{3.52}$$

Dimana nilai eigen value dari (A-LC)

$$eig(A - LC) = [\lambda_1 \ \lambda_2 \ \lambda_3] \tag{3.53}$$

Selisih antara nilai keluaran Observer dengan kecepatan actual motor akan digunakan untuk menentukan nilai penguatan switching pada SMC sehingga menghasilkan persamaan

$$U_{SW} = (K_{observer} + Ks)\tanh\left(\frac{S}{\phi}\right) \tag{3.54}$$

3.5. Aturan Pengujian Sistem

Pengujian dilakukan untuk mengetahui kinerja sistem serta memastikan bahwa hasil perancangan sesuai dengan yang diinginkan. Pengujian yang dilakukan adalah sebagai berikut :

3.5.1. Pengujian model Brushless DC Motor nonlinier dengan loop terbuka dan tertutup.

Pengujian dilakukan pada pemodelan bentuk statespace yang ditambahkan factor ketidak linieran sistem berupa friction column dimana persamaan tersebut

disusun pada blok Simulink pada MATLAB. Input kecepatan berupa sinyal step dengan step time = 1.0 , initial value = 0.0 , dan final value = 100 dalam rad/s.

Pengujian dilakukan bertujuan untuk mengetahui respon transien dari pemodelan motor BLDC saat loop terbuka dan juga loop tertutup. Pengujian ini juga digunakan untuk memastikan bahwa pemodelan yang dilakukan sesuai dengan karakteristik sistem aslinya.

3.5.2. Pengujian pengendali kecepatan dengan *Proportional Integral Derivative* (PID) dan *Sliding Mode Controller* dengan *PID sliding surface*.

Pengujian dilakukan dalam beberapa kondisi sistem. Pertama pengendali PID dan SMC-PID akan di uji pada sistem motor BLDC tidak linier tanpa adanya disturbance atau gangguan. Sistem akan diberi sinyal referensi berupa sinyal step dengan step time = 1, dan final value = 100 rad/sec. Kemudian pada pengujian selanjutnya kedua pengendali tersebut akan diuji dengan gangguan berupa beban sebesar 0.5 N pada saat $t = 2$ sec. Pengujian terakhir dilakukan dengan gangguan tambahan yaitu berupa noise dimana noise tersebut akan muncul pada saat $t = 3$. Kedua pengendali akan diuji respon transien dan steady state-nya dengan 3 kondisi pengujian tersebut. Nilai gangguan dan beban disesuaikan dengan karakteristik motor yang nilainya tidak lebih 40% dari kemampuan sistem.

Tujuan dari pengujian ini adalah untuk mengetahui respon sistem ketika diberi pengendali kecepatan berupa pengendali PID dan SMC-PID dengan berbagai kondisi sistem. Pengujian ini juga bertujuan untuk membandingkan pengendali mana yang memiliki kehandalan dalam mengatasi kondisi misbehavior atau kondisi dimana sistem mengalami gangguan tertentu.

3.5.3. Pengujian pengendali kecepatan dengan *Sliding Mode Controller* dengan *PID sliding surface* berbasis Algoritma Genetika.

Pengujian dilakukan dengan gangguan pada sistem berupa beban dan noise yang besarnya tidak lebih dari 40% dari kemampuan sistem. Beban yang diberikan adalah 0.5 Nm pada saat $t = 2$ sec dan noise pada saat $t = 3$ sec. Step

input yang diberikan sebagai sinyal referensi bernilai 100 rad/sec dengan step time pada saat $t = 1$ sec.

Beberapa parameter pengaturan pada SMC-PID telah ditetapkan dengan metode trial and error diantaranya konstanta switching gain pada SMC sebesar 28, nilai K_p , K_i , dan K_d pada SMC-PID tanpa optimasi masing –masing sebesar 69.5, 80 dan 0.995. Optimasi nilai parameter K_p , K_i , K_d pada PID sliding surface dengan algoritma genetika juga telah ditentukan batas bawah dan batas atasnya sehingga optimasi bisa dilakukan secara maksimal. Batas tersebut bernilai $K_p = [50 \ 80]$, $K_i = [70 \ 100]$ dan $K_d = [0 \ 5]$.

Pengujian dilakukan untuk mendapatkan nilai parameter terbaik pada pengaturan SMC-PID dengan menggunakan metode algoritma genetika (GA). Pengujian ini juga dilakukan untuk membandingkan performa SMC-PID sebelum dan setelah dilakukan optimasi.

